REMARKS/ARGUMENTS

This paper is in response to the Final Office Action dated December 31, 2003. Claims 1-9 and 11-20 are pending in the present application.

35 U.S.C. § 102 Rejections

The Examiner rejected claims 1, 4-6, 9, 11-15, and 18-20 under 35 U.S.C. §102(e) as being anticipated by Mizoguchi et al. (U.S. Patent No. 6,243,858). In so doing, the Examiner stated:

> Referring to claims 1 and 18-20, Mizoguchi discloses a tool for graphically defining an expression with a graphic user interface (GUI) component with means for responding to user input for generating a graphic definition of the expression by defining plurality of tree structures (Figures 6 and 11), wherein as seen by Figure 11, the tree structures represent a distinct nodes under which are represented items which are defined by the user (reference numbers S52 and S53, Figure 10). As also seen by the structures of Figure 11, wherein "input data 1" comprise a hierarchical series of nodes the nodes being represented under "input data 1" and "output data 1", thereby showing a hierarchical relationship. These tree structures also contain a plurality of times listed with the list item being associated with the respective nodes of the associated tree, wherein the nodes and the items both represented as the same data, thereby showing an association are disclosed as being associated with the respective input or output trees. Figure 11 of Mizoguchi also clearly discloses an input data structure and at least one other tree structure representing an output data structure wherein any associated list item defines a formatting definition, used for the graphic expression. Mizoguchi discloses an expression generator component adapted to read the graphic definition of the expression provided by a user through the GUI component, with the expression generator analyzing the graphic definition and generating an expression based on the structure of each tree and any list items associated with respective nodes of a tree (Figure 23), wherein the grid representation which is the graphic definition is executed and the result of the execution is outputted, this involving analyzing and generating of an expression from the graphic representation.

Applicants respectfully disagree. The present invention is directed to a method and tool for graphically defining an expression, in particular, a query containing complex or non-native data types. In the prior art, graphical user interface tools allow a developer to formulate

expressions for configuring either filters for incoming messages (in a messaging and queuing system) or database queries. Those tools, however, only allow users to define values for simple data types.

According to the preferred embodiment of the present invention, the tool includes a graphic user interface (GUI) that allows a user to define a plurality of tree structures comprising a hierarchical series of nodes. At least one of the tree structures represents an input data structure and at least one other tree structure represents an output data structure. The user also is allowed to define one or more lists that include a number of items, whereby each item on a list is associated with a node on a tree. Any item associated with the tree structure representing the input data structure is a filtering constraint and any item associated with the tree structure representing the output data structure defines a formatting definition.

Once the user has defined the plurality of tree structures, an expression generator analyzes the tree/list structure and generates an expression. The expression is used to configure modules in a relational message broker or to configure a database query.

The present invention, as recited in claims 1 and 18, provide:

> 1. A tool for graphically defining an expression, said tool comprising:
> a graphic user interface (GUI) component comprising:
> means, responsive to user input, for generating a graphic definition of the expression by defining a plurality of tree structures comprising a hierarchical series of nodes, and one or more lists comprising a plurality of items, each list item being associated with a respective node of an associated tree structure, wherein at least one of the tree structures represents an input data structure and at least one other tree structure represents an output data structure wherein any associated list item defines a formatting definition;
> an expression generator component adapted to read the graphic definition of the expression provided by a user through said GUI component, expression generator component comprising:
> means for analyzing said graphic definition and generating an expression based on the structure of each tree and any list items associated with respective nodes of a tree.

18. A method for graphically defining an expression in accordance with a graphic definition comprising the steps of:

(a) defining a plurality of tree structures comprising a hierarchical series of nodes, and one or more lists comprising a plurality of items responsive to user input, each list item being associated with a respective node of an associated tree structure, wherein at least one of the tree structures represents an input data structure and at least one other tree structure represents an output data structure wherein any associated list item defines a formatting definition;

(b) analyzing said definition; and

(c) generating an expression based on the structure of each tree and any list items associated with respective nodes of a tree.

Claims 19 and 20 are computer product and system claims, respectively, having similar scopes to that of claim 18.

In contrast to the present invention, Mizoguchi is directed to a tool that assists a software developer in creating a business application program. In Mizoguchi, various basic program modules constituting a business application program are prepared beforehand and made available when the tool is utilized. When an application program is actually constructed, module tiles representing the individual program modules are displayed in the form of a module palette. At the same time, a grid panel on which the module tiles are arranged is displayed. The user arranges the module tiles necessary for the business application program on the grid panel and the tool builds the application program from the user's arrangement. (Abstract).

Mizoguchi clearly fails to teach or suggest generating a graphic definition of an expression by "defining a *plurality of tree structures* comprising a hierarchical series of nodes, and one or more lists comprising a plurality of items, each list item being associated with a respective node of an associated tree structure, wherein at least one of the tree structures represents an input data structure and at least one other tree structure represents an output data structure wherein any associated list item defines a formatting definition," as recited in claims 1, 18-20. In Mizoguchi, the tool allows the user to connect a chain of program modules to create a graphic representation of an application program. Each program module performs various types

of data processing, for example update processing, retrieval processing, data manipulation processing, input processing and print processing. (Col. 7, lines 39-50). Only one chain of modules is created. (See for example, Figures 6A-6C, 8, 13 and 25). Accordingly, Mizoguchi teaches defining *one* chain or tree structure representing an application program. Mizoguchi does not teach or suggest defining "a *plurality* of tree structures," as recited in claims 1, and 18-20, and certainly does not teach or suggest defining tree structures having the properties recited in claims 1 and 18-20.

In the Final Office Action, the Examiner indicates that the plurality of tree structures is taught in Figures 6 and 11 of Mizoguchi, and that "the tree structures represent a distinct nodes under which are represented items which are defined by the user" (Figure 11). If the Examiner interprets each program module (tile) to be a distinct node in a tree structure, then (as stated above) Mizoguchi teaches defining only one tree structure and not *a plurality* of tree structures, as recited in independent claims 1, and 18-20.

If, on the other hand, the Examiner interprets each program module to be a "tree structure," then Applicants respectfully submit that the program modules do not comprise "a hierarchical series of nodes," as recited in claims 1, and 18-20. As stated above, the program modules perform various types of data processing (col. 2, lines 1-2; col. 7, lines 39-50). Intuitively, a program module comprises lines of programming code that instruct the module to process input data according to the module's purpose. Nothing teaches or suggests that any of the program modules is a tree structure "comprising a hierarchical series of nodes," as recited in claims 1, and 18-20.

If, alternatively, the Examiner interprets each program module comprising a plurality of tree structures represented (in Figure 11) by the input data record list, the processing list, and the output data record item list respectively (as shown in item 60 of Figure 11), then Applicants

again respectfully submit that such lists fail to teach or suggest "a plurality of tree structures comprising a hierarchical series of nodes," as recited in claims 1, and 18-20.

Figure 11 depicts the "data record processing definition screen 60" associated with the "retrieval processing" program module. This screen 60 is displayed to the user after he or she has completed defining input and output *data records* for the module (steps S4 and S5 of Figure 5). In the definition screen 60, "the input data record items are listed in a processing definition frame of input data records and the output data record items are listed in a processing definition frame of output data records. The description of the processing of output data record items by reference to the input data record items can be written in the PROCESSING area." (Col. 10, lines 1-7).

Applicants respectfully submit that Figure 11 does not teach or suggest one tree structure or a plurality of tree structures "comprising a *hierarchical* series of nodes" but rather, it displays lists of input and output data record items and what processing, if any, was performed on each input record item to produce the corresponding output data item. While each item on the input list *is transformed* to a corresponding item on the output list (via the processing), nothing teaches or suggests that the relationship between the items on the input list and the items on the output list is *hierarchical*, or that the relationship *between* the items on any particular list is hierarchical. For example, referring to Figure 11, *output* data item 2 is related mathematically to *input* data item 2 by the function, "input data item 2+100." This is *not* a hierarchical relationship. Also, nothing in Mizoguchi teaches or suggests any relationship, let alone a hierarchical relationship, between record items on the input or output lists.

In addition, nothing in Mizoguchi in general and in particular Figure 11 teaches or suggests "one or more lists comprising a plurality of items, *each list item being associated with a respective node of an associated tree*," as recited in claims 1, 18-20. If the program module is

represented by a tree structure and each list in the definition screen 60 is considered a node (which is disputed above), then, contrary to the present invention, each node is associated with *several* items on a list.

Finally, nothing in Mizoguchi in general and in particular Figure 11 teaches or suggests "an output data structure wherein any associated list item defines *a formatting definition*," as recited in claims 1 and 18-20. In the Final Office Action, the Examiner again relies on Figure 11 to teach this feature. Presumably, the Examiner interprets the output data record item list as "an output data structure." Nevertheless, as stated above, the output data record item list fails to teach or suggest a tree structure "comprising a hierarchical series of nodes." Even if it did, nothing teaches or suggests that output data record item on the list "defines a formatting definition." Indeed, according to Mizoguchi, the definition of the data record means "the *name* of the data record itself . . . and the *names* of the items included in the data record." (Col. 8, lines 53-57). Nothing teaches or suggest that the *name* of the output data record item "defines a formatting definition," as recited in claims 1 and 18-20.

Accordingly, based on the arguments above, Applicants respectfully submit that Mizoguchi fails to teach or suggest the cooperation of elements recited in claims 1, 18, 19 and 20. Thus, claims 1, 18, 19 and 20 are allowable over the cited reference. Claims 4-6, 9, and 11-15 on claim 1, and the above arguments apply with full force. Therefore, claims 4-6, 9, and 11-15 are also allowable over the cited reference.

35 U.S.C. §103 Rejections

The Examiner rejected claims 2, 3 and 16 under 35 U.S.C. §103(a) as being unpatentable over Mizoguchi in view of MacLeod et al (U.S. Patent No. 6,434,545). Claim 7 was rejected under 35 U.S.C. §103(a) as being unpatentable over Mizoguchi and Premerlani et al. (U.S. Patent

No. 5,555,367). Claim 8 was rejected as being unpatentable over Mizoguchi and Lee et al. (U.S. Patent No. 6,535,883). Claim 17 was rejected as being unpatentable over Mizoguchi in view of MacLeod and further in view of Moshfeghi (U.S. Patent No. 6,476,833).

Claims 2, 3, 7, 8, 16 and 17 depend on claim 1 and therefore, the arguments with regard to claim 1 apply with full force to claims 2, 3, 7, 8, 16 and 17. Thus, even if the secondary references disclose the features described by the Examiner, claims 2, 3, 7, 8, 16 and 17 are still allowable because Mizoguchi fails to teach or suggest the present invention as recited in claim 1. Accordingly, Applicants respectfully submit that claims 2, 3, 7, 8, 16 and 17 are allowable over the cited references.

Conclusion

In view of the foregoing, it is submitted that the claims 1-9 and 11-20 are allowable over the cited references and are in condition for allowance. Applicants respectfully request reconsideration of the rejections and objections to the claims, as now presented.

Applicants believe that this application is in condition for allowance. Should any unresolved issues remain, Examiner is invited to call Applicants' attorney at the telephone number indicated below.

Respectfully submitted,
SAWYER LAW GROUP LLP

March 1, 2004
Date

Joyce Tom
Attorneys for Applicant(s)
Reg. No. 48,681
(650) 493-4540